



Deliverable-01

WIS Testing Report

Workgroup: E7.04

Date: 02/03/2022

Miembros:

Baños González, Alejandro (alebangon@alum.us.es)

Flores Rodríguez, Román (romflorod@alum.us.es)

Grosso Gómez de Terreros, Javier (javgrogom@alum.us.es)

Gutiérrez Ceballos, Pablo (pabgutceb@alum.us.es)

Ibáñez Montero, Julia (julibamon@alum.us.es)

Roldán Cadena, Jesús (jesroldcad@alum.us.es)

Repositorio de Github: <https://github.com/jesroldcad/Acme-Toolkits>

Índice

1. Resumen ejecutivo	1
2. Tabla de control de cambios	2
3. Introducción	3
4. Testing de un Proyecto Software	4
5. Experiencia previa del equipo	5
5.1 Pruebas unitarias	5
5.2. Dobles de pruebas	5
6. Conclusiones	7
7. Referencias bibliográficas	8

1. Resumen ejecutivo

En este documento se detallan los conocimientos base con los que cuenta el equipo de desarrollo del proyecto en relación al testing de software.

A pesar de que todos los miembros del equipo hemos trabajado diversas formas y técnicas de testing, nos centraremos en mencionar y ahondaremos en aquellas que conocemos en mayor medida y con las cuales nos sentimos más cómodos y eficientes.

Dicho esto, y tras haber consensuado que el conocimiento general del grupo en cuanto al testing software parte de los mismos cimientos, se tratarán principalmente las pruebas unitarias y las pruebas con dobles de pruebas, si bien mencionaremos otras técnicas que también conocemos, pero en las cuales no profundizaremos tanto.

En conclusión, creemos que, aunque nos queda mucho por aprender, tenemos una base general suficiente sobre testing para poder enfrentarnos al nuevo proyecto que nos brinda la asignatura Diseño y Pruebas II.

2. Tabla de control de cambios

Número de revisión	Descripción	Fecha
1	Creación del documento acorde a la plantilla.	26/02/2022
2	Completados todos los apartados del documento.	26/02/2022
3	Revisión general del documento. Se han realizado pequeñas correcciones de forma y contenido.	27/02/2022

3. Introducción

En este documento se va a reflejar la experiencia y conocimiento que tiene el equipo en relación con las técnicas de testeo de software, así como una breve descripción de las mismas. Para ello, hemos organizado este documento en distintas secciones descritas a continuación.

Sección 4. *Testing de un Proyecto Software.* Antes de hablar de las técnicas con las que estamos más familiarizados, se hará una breve descripción de qué es el testing software y la importancia que tiene en el desarrollo de un proyecto.

Sección 5. *Experiencia previa del equipo (5.1 Pruebas unitarias, 5.2 Dobles de pruebas).* En esta sección se hará una breve descripción de la base de conocimiento desde la que parte el equipo en cuanto a testing. En cada uno de los subapartados se van a describir de manera breve cada una de ellas.

Sección 6. *Conclusiones.* Se reflejarán las conclusiones que hemos compartido todo el equipo acerca de los conocimientos previos que tenemos en común y cómo pensamos que nos vamos a enfrentar a la asignatura Diseño y Pruebas II.

Sección 7. *Referencias bibliográficas.* En esta sección reflejaremos las fuentes que hemos utilizado para recopilar alguna información necesaria para el correcto desarrollo de este informe.

Finalmente, el documento cuenta con la siguiente estructura: portada, índice, resumen ejecutivo, tabla de control de cambios, introducción, contenidos (divididos en las secciones *Testing de un Proyecto Software* y *Experiencia previa del equipo*, esta sección a su vez dividida en las subsecciones *Pruebas unitarias* y *Dobles de pruebas*), conclusiones y referencias bibliográficas.

4. Testing de un Proyecto Software

El testing de software es un proceso para verificar y validar la funcionalidad de un programa o una aplicación de software con el objetivo de garantizar que el producto de software esté libre de defectos, y que cumpla con los requisitos esperados, para poder entregar un producto de calidad. Implica la ejecución de componentes de software o sistema utilizando herramientas manuales o automatizadas para evaluar una o más propiedades de interés.

El testing de software es un proceso paralelo al desarrollo de software (el testing debe ir realizándose a medida que se construye el producto para evitar problemas en la funcionalidad de manera previa a su lanzamiento).

Las pruebas de software son importantes porque permiten identificar de manera temprana si hay algún problema en el software, facilitando su resolución antes de la entrega del producto.

Se clasifican generalmente en dos categorías principales:

- Pruebas funcionales: Basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software (requisitos funcionales).
- Pruebas no funcionales o pruebas de rendimiento: Consideran parámetros como la confiabilidad, la usabilidad y el rendimiento.

Las pruebas, además, también se pueden clasificar en pruebas manuales o automáticas.

5. Experiencia previa del equipo

Todos los miembros del equipo hemos estudiado qué es el testing de un proyecto software, así como los distintos tipos de técnicas y pruebas que lo conforman.

Sin embargo, vamos a profundizar en aquellas pruebas de software con las cuales estamos más familiarizados ya que las hemos trabajado tanto de manera teórica, pero sobre todo y más importante de manera práctica.

5.1 Pruebas unitarias

Pertenecen a la categoría de las denominadas pruebas funcionales.

En programación, una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Esto sirve para asegurar que cada unidad funcione correctamente y eficientemente por separado. Además de verificar que el código hace lo que tiene que hacer, verificamos que sea correcto el nombre, los nombres y tipos de los parámetros, el tipo de lo que se devuelve, que si el estado inicial es válido, entonces el estado final es válido también.

5.2. Dobles de pruebas

Para explicar lo que es un doble de prueba, vamos a centrarnos inicialmente en las pruebas unitarias automatizadas. Las pruebas unitarias están ideadas para verificar que cada pequeña pieza de código que constituye un sistema, hace lo que tiene que hacer.

Los sistemas están formados por cientos o miles de componentes que interactúan entre sí. Gracias a ello dan soporte a las funcionalidades del sistema. Por ello al probar un componente estamos probando el propio componente y todos con los que interactúa (no es el objetivo intrínseco de una prueba unitaria).

El concepto dobles de prueba nace con el objetivo de aislar un componente del resto de componentes con los que interactúa, para, valga la redundancia, probarlo de manera aislada.

Con el objetivo de aislar un componente del resto de los componentes con los que interactúa para probarla de forma aislada, se ideó el concepto de dobles de test. Un doble de test es una clase específica para pruebas que sustituye a una pieza real que interacciona con la pieza que se está probando.

6. Conclusiones

Tras haber puesto en común la redacción y posterior lectura de este documento, los integrantes del equipo de desarrollo hemos sacado varias conclusiones relacionadas con nuestra experiencia en el testing software y cómo enfrentarnos al nuevo proyecto que tenemos sobre la mesa.

Si bien no somos expertos en el testeo de un proyecto software, consideramos que nuestra base teórica es bastante sólida como para poder abordar con determinación el proyecto de la asignatura.

Por otro lado, estamos al tanto de que el testeo es una parte tediosa del desarrollo del proyecto y que quizás no tenemos toda la experiencia práctica que nos gustaría.

Somos conscientes de que el testing, al ser algo tan importante para el proyecto, nos vamos a tener que involucrar todos para poder entregar un producto final que cumpla con unos mínimos de calidad exigidos.

7. Referencias bibliográficas

-Sección 4 y 5:

Qué es el testing software

<https://profile.es/blog/que-es-el-testing-de-software/>

Pruebas de software

https://es.wikipedia.org/wiki/Pruebas_de_software#Definici%C3%B3n

Dobles de pruebas

<http://www.asuntosoftware.com/2019/09/pruebas-software-dobles-de-prueba-con.html>